UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/723,051	11/25/2003	Ernest P. Chen	P17529	7780

25694 7590 02/22/2006

INTEL CORPORATION
P.O. BOX 5326
SANTA CLARA, CA 95056-5326

EXAMINER	
GEIB, BENJAMIN P	

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 02/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/723,051	CHEN, ERNEST P.	
	Examiner	Art Unit	
	Benjamin P. Geib	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 November 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-56* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 November 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-56 have been examined.

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Application on 11/25/2003.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 24-29 and 44-49 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The listed claims recite the limitation of a "machine-readable medium" which is defined in the specification *(last paragraph on page 3 and continuing on page 4)* to include the following non-statutory subject matter: "electrical, optical, acoustical, or other forms of propagated signals (e.g. carrier waves, infrared signals, digital signals, etc.)".

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-56 are rejected under 35 U.S.C. 102(b) as being anticipated by Turley, "Advanced 80386 Programming Techniques" (Herein referred to as Turley).

6. Referring to claim 1, Turley has taught a method comprising:

setting a first indicator *[trace flag (Fig. 10-1), breakpoint opcode, and global enable bit (Fig. 10-3)]*; and

interrupting execution of a computer program instruction in response to setting the first indicator *[The following are three separate instances (i.e. interpretations) of setting a first indicator and interrupting execution of a computer program instruction in response to setting the first indicator:*

1) setting the trace flag (See second paragraph on page 324 and Fig. 10-1)

2) exchanging an instruction byte with the breakpoint opcode 0xCC (See second and third paragraph on page 326)

3) setting a global enable bit (see page 333-334)]

7. Referring to claim 2, Turley has taught the method of claim 1, further comprising:

setting a second indicator *(exchanging an instruction byte with the breakpoint opcode 0xCC)*; and

halting execution of the computer program instruction in response to setting the first and second indicators *(See second and third paragraphs on page 326)*.

8. Referring to claim 3, Turley has taught the method of claim 2, further comprising:

resetting the first indicator *(The IRET instruction of the single step handler resets the trace flag; See last paragraph on page 324)*; and

resuming execution of the computer program instruction in response to resetting the first indicator *(In response to resetting the trace flag with the IRET instruction the execution of the computer program instruction is resumed since, as is well known in the art, the IRET instruction returns execution to the interrupted instruction; See last paragraph on page 324)*.

9. Referring to claim 4, Turley has taught the method of claim 2, wherein the first and second indicators comprise data bits *(See first full paragraph on page 324 and the third paragraph on page 326)*.

10. Referring to claim 5, Turley has taught the method of claim 4, wherein the data bits are held in a register *[The trace flag is stored in the EFLAGS register (See first full paragraph on page 324) and the breakpoint opcode is held in a memory location, which is a register (See fourth paragraph on page 326)]*.

11. Referring to claim 6, Turley has taught the method of claim 1, further comprising:

setting a second indicator *(setting a global enable bit (see Fig. 10-3))*;

setting a third indicator *(setting the trace flag (See second paragraph on page 324 and Fig. 10-1))*; and

halting execution of the computer program instruction in response to setting the second and third indicators *[Execution of the instruction is halted in response to setting the global enable bit (see page 334). Execution of the instruction is also halted in response to setting the trace flag (See second paragraph on page 324)]*.

12. Referring to claim 7, Turley has taught the method of claim 6, further comprising:

resetting the third indicator *(The IRET instruction of the single step handler resets the trace flag; See last paragraph on page 324)*; and

resuming execution of the computer program instruction in response to resetting the third indicator *(In response to resetting the trace flag with the IRET instruction the execution of the computer program instruction is resumed since, as is well known in the art, the IRET instruction returns execution to the interrupted instruction; See last paragraph on page 324)*.

13. Referring to claim 8, Turley has taught the method of claim 6, wherein the computer program instruction includes the first indicator *(The breakpoint opcode (i.e. first indicator) is included in the computer program instruction; See second and third paragraphs on page 326)*.

14. Referring to claim 9, Turley has taught the method of claim 6, wherein the second and third indicators comprise data bits *(See second paragraph on page 324 and second paragraph on page 334)*.

15. Referring to claim 10, Turley has taught the method of claim 9, wherein the data bits are held in a register *(The trace flag is stored in the EFLAGS register and the global enable is stored in a Debug register; See second paragraph on page 324 and second paragraph on page 334)*.

16. Referring to claim 11, Turley has taught the method of claim 1, further comprising:

resetting the first indicator *[The IRET instruction of the single step handler resets the trace flag (i.e. first indicator) (See last paragraph on page 324)]*;

setting a second indicator *[Exchanging the first byte of instructions with the breakpoint opcode 0xCC (See third paragraph on page 326)]*; and

halting execution of the computer program instruction in response to resetting the first indicator *(The trace flag, after being reset by the IRET instruction, continues to halt execution of the instruction; See last paragraph on page 324)* and setting the second indicator *(See second and third paragraphs on page 326)*.

17. Referring to claim 12, Turley has taught the method of claim 11, further comprising:

resetting the second indicator *[The breakpoint opcode is swapped with the original byte of the interrupted instruction and inserted into a subsequent instruction (i.e. the indicator is reset) (See first and second paragraphs on page 327)]*; and

resuming execution of the computer program instruction in response to resetting the second indicator *[After resetting the breakpoint opcode as stated above, execution of the computer program instruction is resumed (See first and second paragraphs on page 327)]*.

18. Referring to claim 13, Turley has taught the method of claim 11, further comprising:

setting the first indicator *[As previously noted (in the arguments to claim 1) setting the trace flag (See second paragraph on page 324 and Fig. 10-1)]*;

resetting the second indicator *[The breakpoint opcode is swapped with the original byte of the interrupted instruction and inserted into a subsequent instruction (i.e. the indicator is reset) (See first and second paragraphs on page 327)]*;

resuming execution of the computer program instruction in response to setting the first indicator *(In response to setting the trace flag with the IRET instruction the execution of the computer program instruction is resumed since, as is well known in the art, the IRET instruction returns execution to the interrupted instruction; See last paragraph on page 324)* and to resetting the second indicator *[After resetting the breakpoint opcode as stated above, execution of the computer program instruction is resumed (See first and second paragraphs on page 327)]*; and

interrupting execution of a subsequent computer program instruction in response to setting the first indicator *[Execution of subsequent instructions is also interrupted in response to setting the trace flag (See second paragraph on page 324)]* and to resetting the second indicator *[After resetting the breakpoint opcode as stated above, execution of the subsequent instruction is halted (See first and second paragraphs on page 327)]*.

19. Referring to claim 14, Turley has taught the method of claim 13, further comprising:

resetting the first indicator *[As previously noted (in the arguments to claim 11), the IRET instruction of the single step handler resets the trace flag (i.e. first indicator) (See last paragraph on page 324)]*;

setting the second indicator *[As previously noted (in the arguments to claim 11), exchanging the first byte of instructions with the breakpoint opcode 0xCC (See third paragraph on page 326)]*; and

halting execution of the subsequent computer program instruction in response to resetting the first indicator *[Execution of subsequent instructions is also interrupted in*

response to setting the trace flag (See second paragraph on page 324)] and to setting the second indicator *[Since the subsequent instruction includes the breakpoint opcode, execution of the subsequent instruction is halted (See first and second paragraphs on page 327)].*

20. Referring to claim 15, Turley has taught the method of claim 11, wherein the first and second indicators comprise data bits *(See second paragraph on page 324 and third paragraph on page 326).*

21. Referring to claim 16, Turley has taught the method of claim 15, wherein the data bits are held in a register *[The trace flag is stored in the EFLAGS register (See second paragraph on page 324) The breakpoint opcode is held in a memory location, which is a register (See fourth paragraph on page 326)].*

22. Referring to claim 17, Turley has taught an apparatus comprising:

a processor *(80386 microprocessor)*; and

at least one indicator coupled to the processor *(an instruction byte with the breakpoint opcode 0xCC)*, wherein the at least one indicator is configurable to halt execution of a computer program instruction by the processor *(See second and third paragraphs on page 326).*

23. Referring to claim 18, Turley has taught the apparatus of claim 17, wherein the at least one indicator comprises at least one data bit readable by the processor *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326).*

24. Referring to claim 19, Turley has taught the apparatus of claim 18, wherein the at least one data bit is held in a register coupled to the processor *(The breakpoint opcode*

is held in a memory location, which is a register, coupled to the processor; See fourth paragraph on page 326).

25. Referring to claim 20, Turley has taught the apparatus of claim 17, wherein the computer program instruction executed by the processor includes at least one indicator configurable to interrupt execution of the computer program instruction *(The breakpoint opcode indicator is included in the computer program instruction; See second and third paragraphs on page 326).*

26. Referring to claim 21, Turley has taught the apparatus of claim 20, wherein the at least one indicator included in the computer instruction comprises at least one data bit *(The breakpoint opcode is held in a memory location, which is a register, coupled to the processor; See fourth paragraph on page 326).*

27. Referring to claim 22, Turley has taught the apparatus of claim 17 wherein the at least one indicator is reconfigurable to resume execution of the computer program instruction *[The breakpoint opcode is swapped with the original byte of the interrupted instruction (i.e. the indicator is reconfigured) and execution of the computer program instruction is resumed; See first full paragraph on page 327].*

28. Referring to claim 23, Turley has taught the apparatus of claim 17 wherein the at least one indicator is reconfigurable to resume execution of the computer program instruction and to halt execution of a subsequent computer program by the processor *[The breakpoint opcode is swapped with the original byte of the interrupted instruction and inserted into a subsequent instruction (i.e. the indicator is reconfigured) and*

execution of the computer program instruction is resumed and execution of the subsequent instruction is halted; See first and second paragraphs on page 327].

29. Referring to claim 24, Turley has taught a machine-readable medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

configuring at least one indicator coupled to a processor *(Exchanging an instruction byte with the breakpoint opcode 0xCC; See third paragraph on page 326)*;

halting execution by the processor of an instruction issued by a computer program in response to the configuring of the at least one indicator *(See second and third paragraphs on page 326)*;

reconfiguring the at least one indicator *(The breakpoint opcode is swapped with the original byte of the interrupted instruction)*; and

finishing execution by the processor of the instruction issued by the computer program in response to the reconfiguring of the at least one indicator *(Execution of the instruction is resumed after replacing the breakpoint opcode with the original instruction byte; See first full paragraph on page 327).*

30. Referring to claim 25, Turley has taught the machine-readable medium of claim 24, wherein the at least one indicator comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326).*

31. Referring to claim 26, Turley has taught the machine-readable medium of claim 25, wherein the at least one data bit is held in a register coupled to the processor *(The*

breakpoint opcode is held in a memory location, which is a register, coupled to the processor; See fourth paragraph on page 326).

32. Referring to claim 27, Turley has taught the machine-readable medium of claim 24, wherein the at least one indicator comprises at least one indicator included in the computer program instruction, wherein the at least one indicator included in the computer program instruction is configurable to interrupt execution of the computer program instruction *(The breakpoint opcode indicator is included in the computer program instruction and interrupts the execution of the instruction; See second and third paragraphs on page 326).*

33. Referring to claim 28, Turley has taught the machine-readable medium of claim 27, wherein the at least one indicator included in the computer program instruction comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326).*

34. Referring to claim 29, Turley has taught the machine-readable medium of claim 24, wherein reconfiguring the at least one indicator causes the processor to halt execution of a subsequent computer program instruction *(The breakpoint opcode is inserted into a subsequent instruction and execution of the subsequent instruction is halted; See first and second paragraphs on page 327).*

35. Referring to claim 30, Turley has taught a system comprising:

a processor to execute computer program instructions *(80386 microprocessor; See page 326)*;

a memory coupled to the processor, the memory to store the computer program instructions to be executed by the processor *(See page 326)*; and

at least one indicator coupled to the processor *(an instruction byte with the breakpoint opcode 0xCC)*, the at least one indicator configurable to control execution of the computer program instructions by the processor *(See second and third paragraphs on page 326)*.

36. Referring to claim 31, Turley has taught the system of claim 30, wherein the at least one indicator comprise at least one data bit configurable to halt execution of one or more of the computer program instructions by the processor *(The breakpoint opcode indicator comprises 8 bits and halts execution of an instruction; See third paragraph on page 326)*.

37. Referring to claim 32, Turley has taught the system of claim 31, wherein the at least one data bit is reconfigurable to resume execution of the one or more of the computer program instructions by the processor *(The breakpoint opcode is swapped with the original byte of the interrupted instruction and execution of the instruction is resumed; See first full paragraph on page 327)*.

38. Referring to claim 33, Turley has taught the system of claim 30, wherein the at least one indicator comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326)*.

39. Referring to claim 34, Turley has taught the system of claim 33, wherein the at least one data bit is held in a register *(The breakpoint opcode is held in a memory*

location, which is a register, coupled to the processor; See fourth paragraph on page 326).

40. Referring to claim 35, Turley has taught the system of claim 30, wherein the at least one indicator comprises at least one indicator included in the computer program instruction, wherein the at least one indicator included in the computer program instruction is configurable to interrupt execution of the computer program instruction *(The breakpoint opcode indicator is included in the computer program instruction and interrupts the execution of the instruction; See second and third paragraphs on page 326).*

41. Referring to claim 36, Turley has taught the system of claim 35, wherein the at least one indicator included in the computer program instruction comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326).*

42. Referring to claim 37, Turley has taught an apparatus comprising:

at least one indicator configurable to halt execution of a computer program instruction by embedded logic *(an instruction byte with the breakpoint opcode 0xCC; See second and third paragraphs on page 326)..*

43. Referring to claim 38, Turley has taught the apparatus of claim 37, wherein the at least one indicator is at least one data bit readable by embedded logic *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326).*

44. Referring to claim 39, Turley has taught the apparatus of claim 38, wherein the at least one data bit is held in a register *(The breakpoint opcode is held in a memory*

location, which is a register, coupled to the processor; See fourth paragraph on page 326).

45. Referring to claim 40, Turley has taught the apparatus of claim 37, wherein the computer program instruction executable by embedded logic includes at least one indicator configurable to interrupt execution of the computer program instruction *(The breakpoint opcode indicator is included in the computer program instruction and interrupts the execution of the instruction; See second and third paragraphs on page 326).*

46. Referring to claim 41, Turley has taught the apparatus of claim 40, wherein the at least one indicator included in the computer instruction comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326).*

47. Referring to claim 42, Turley has taught the apparatus of claim 37, wherein the at least one indicator is reconfigurable to resume execution of the computer program instruction *(The breakpoint opcode is swapped with the original byte of the interrupted instruction and execution of the instruction is resumed; See first full paragraph on page 327).*

48. Referring to claim 43, Turley has taught the apparatus of claim 37, wherein the at least one indicator is reconfigurable to resume execution of the computer program instruction and to halt execution of a subsequent computer program *[The breakpoint opcode is swapped with the original byte of the interrupted instruction and inserted into a subsequent instruction (i.e. the indicator is reconfigured) and execution of the*

computer program instruction is resumed and execution of the subsequent instruction is halted; See first and second paragraphs on page 327].

49. Referring to claim 44, Turley has taught a machine-readable medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

configuring at least one indicator readable by embedded logic *(Exchanging an instruction byte with the breakpoint opcode 0xCC; See third paragraph on page 326)*;

halting execution by embedded logic of an instruction issued by a computer program in response to the configuring of the at least one indicator *(See second and third paragraphs on page 326)*;

reconfiguring the at least one indicator *(The breakpoint opcode is swapped with the original byte of the interrupted instruction)*; and

finishing execution by embedded logic of the computer program instruction in response to the reconfiguring of the at least one indicator *(Execution of the instruction is resumed after replacing the breakpoint opcode with the original instruction byte; See first full paragraph on page 327)*.

50. Referring to claim 45, Turley has taught the machine-readable medium of claim 44, wherein the at least one indicator comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326)*.

51. Referring to claim 46, Turley has taught the machine-readable medium of claim 45, wherein the at least one data bit is held in a register *(The breakpoint opcode is held*

in a memory location, which is a register, coupled to the processor; See fourth paragraph on page 326).

52. Referring to claim 47, Turley has taught the machine-readable medium of claim 44, wherein the at least one indicator comprises at least one indicator included in the computer program instruction, wherein the at least one indicator included in the computer program instruction is configurable to interrupt execution of the instruction *(The breakpoint opcode indicator is included in the computer program instruction and interrupts the execution of the instruction; See second and third paragraphs on page 326).*

53. Referring to claim 48, Turley has taught the machine-readable medium of claim 47, wherein the at least one indicator included in the computer program instruction comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326).*

54. Referring to claim 49, Turley has taught the machine-readable medium of claim 44, wherein embedded logic halts execution of a subsequent computer program instruction in response to reconfiguring the at least one indicator *(The breakpoint opcode is inserted into a subsequent instruction and execution of the subsequent instruction is halted; See first and second paragraphs on page 327).*

55. Referring to claim 50, Turley has taught a system comprising:

computer program instructions executable by embedded logic *(See page 326)*;

a memory to store the computer program instructions *(See page 326)*; and

at least one indicator *(an instruction byte with the breakpoint opcode 0xCC)* configurable to control execution of the computer program instructions *(See second and third paragraphs on page 326)*.

56. Referring to claim 51, Turley has taught the system of claim 50, wherein the at least one indicator comprises at least one data bit configurable to halt execution of one or more of the computer program instructions *(The breakpoint opcode indicator comprises 8 bits and halts execution of an instruction; See third paragraph on page 326)*.

57. Referring to claim 52, Turley has taught the system of claim 51, wherein the at least one data bit is reconfigurable to resume execution of the one or more of the computer program instructions *(The breakpoint opcode is swapped with the original byte of the interrupted instruction and execution of the instruction is resumed; See first full paragraph on page 327)*.

58. Referring to claim 53, Turley has taught the system of claim 50, wherein the at least one indicator comprises at least one indicator included in the computer program instruction, wherein the at least one indicator included in the computer program instruction is configurable to interrupt execution of the instruction *(The breakpoint opcode indicator is included in the computer program instruction and interrupts the execution of the instruction; See second and third paragraphs on page 326)*.

59. Referring to claim 54, Turley has taught the system of claim 53, wherein the at least one indicator included in the computer program instruction comprises at least one

data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326)*.

60. Referring to claim 55, Turley has taught the system of claim 50, wherein at least one indicator comprises at least one data bit *(The breakpoint opcode indicator comprises 8 bits; See third paragraph on page 326)*.

61. Referring to claim 56, Turley has taught the system of claim 55, wherein the at least one data bit is held in a register *(The breakpoint opcode is held in a memory location, which is a register, coupled to the processor; See fourth paragraph on page 326)*.

Conclusion

62. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

63. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Millas et al., "Breakpoint Bit Technique for an Address Compare Function", teaches a method of breakpointing where a bit is included with the instruction indicating whether or not to breakpoint at the instruction.

Key et al., U.S. Patent No. 6,173,386, teaches a system including a host processor running a debugger and a parallel processor running debuggee code.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Benjamin P. Geib whose telephone number is (571) 272-8628. The examiner can normally be reached on Mon-Fri 8:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Huynh can be reached on (571) 272-4147. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Benjamin P Geib
Examiner
Art Unit 2181

2/17/06

HENRY W. H. TSAI
PRIMARY EXAMINER